

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the instant application:

Listing of Claims:

1. (Previously Presented) A computer-implemented method for providing dynamic workload transition in an application server for an e-business system, comprising:

receiving at least one work request that includes at least one workload task having an associated workload parameter specifying a resource requirement;

determining a resource capacity of the e-business system in view of said resource requirement;

identifying a priority of said workload task over workload tasks currently executing within said e-business system;

predicting an overload condition in view of said resource capacity for at least one system having priority in the e-business system for executing a portion of said workload task;

causing a first reallocation of at least a portion of system resources allocated to a first set of workload tasks in the e-business system from said first set of workload tasks to a second set of workload tasks in response to predicting the overload condition,

executing a query of at least a portion of said first set of workload tasks included in said workload request in response to said first reallocation,

wherein processing said second set of workload tasks requires less system resources than processing said first set of workload tasks, and wherein said workload tasks are performed by a plurality of different applications under direction of the e-business system; and

if said overload condition subsequently abates and if said first set of workload tasks require processing, performing a second reallocation of system resources to said first set of workload tasks.

2. (Previously Presented) The computer-implemented method according to claim 1, wherein said detecting step further comprises

receiving said workload request in a text format for providing visualization of a resource requirement;

monitoring system parameters in the e-business system; and

analyzing said monitored system parameters to predict when said overload condition occurs in the e-business system.

3. (Previously Presented) The computer implemented method according to claim 2, wherein said text format is an XML presentation of monitored system parameters comprising CPU utilization, disk I/O and memory utilization.

4. (Previously Presented) A computer-implemented method for providing dynamic workload transition in an application server for an e-business system, comprising:

receiving a first work request for performing a workload task having one or more associated workload parameters specifying a resource requirement by at least one application under direction of the e-business system;

receiving HTTP status updates from one or more systems within the e-business system describing an available resource capacity of the one or more systems based on said workload parameter;

updating a status servlet and a core workload driver within and an application server using said HTTP status updates to identify systems with available processing capacity;

comparing said resource requirement of said first work request to identified available system resources to predict if performing said workload task of said first work request is capable of causing a system overload condition in at least one system within the e-business system; and

if performing said workload task of said first work request is capable of causing a system overload condition, transitioning to a second lighter work request, said second lighter work request for performing a different workload task having a lighter workload requiring less system resources, thereby preventing said system overload condition.

5. (Previously Presented) The computer-implemented method according to claim 4, further comprising analyzing an XML representation of said system parameters received within said HTTP request to predict whether performing said workload task of said first work request causes said system overload condition.

6. (Previously Presented) The computer-implemented method according to claim 5, wherein said system parameters comprises CPU utilization, disk I/O and memory utilization.

7. (Previously Presented) The computer-implemented method according to claim 5, further comprising, reporting said system parameters in an XML format under an HTTP request to a workload driver.

8. (Previously Presented) A computer-implemented method for providing dynamic workload transition in an application server for an e-business system, comprising:

processing a workload task having one or more associated workload parameters specifying one or more resource requirements performed by at least one application under direction of the e-business system, the workload task assigned in an HTTP request to a workload driver;

monitoring e-business system resources in view of said one or more resource requirements;

predicting an overload condition in the e-business system while processing said workload task;

allocating processing resources to a lighter workload task when said workload driver predicts a system overload condition caused by said processed workload task during said monitoring step; and

executing a query of at least a portion of said lighter workload task in response to said first reallocation for acquiring a portion of data requested by said HTTP request,

if said processed workload task still requires processing, reporting a result of said executing a query and responding with a message to make a subsequent request to acquire a remainder of data made available by said processing followed by transitioning to say processed workload task from said lighter workload task upon availability of adequate processing resources.

9. (Currently Amended) A software system implemented on a e-business application server ~~operating on a computer-readable medium~~ for providing and monitoring dynamic workload transition in an e-business system, the system comprising:

~~an application server for receiving HTTP work requests and for processing workload tasks in the e-business system, the workload tasks being identified and visually presented by said work requests and being performed by a plurality of different applications under direction of the e-business system;~~

a workload driver for handling workload management of said one or more HTTP work requests received by said application server, wherein said handling ~~comprising~~ comprises: predicting a resource requirement of one or more systems networked to said application server from at least one parameter specified in said HTTP work request, diminishing processing of a currently processed workload task which causes an overload condition in the e-business system in view of said resource requirement, executing a query for at least a portion of said HTTP work request, and initiating the processing of a lighter workload task, said lighter workload task having a lighter workload than said currently processed workload task; and

a status driver for reporting system resource capacity data in an XML format within an HTTP request to said workload driver, said system resource capacity data providing textual information regarding the existence of said overload condition.

10. (Previously Presented) A machine-readable storage having stored thereon, a computer program having a plurality of code sections, said code sections executable by a machine for causing the machine to perform the steps of:

receiving at least one work request that includes at least one workload task having an associated workload parameter specifying a resource requirement;

determining a resource capacity of the e-business system in view of said resource requirement;

identifying a priority of said workload task over workload tasks currently executing within said e-business system;

predicting an overload condition in view of said resource capacity for at least one system having priority in the e-business system for executing a portion of said workload task; said detecting step for providing dynamic workload transition in an application server for the e-business system;

causing a reallocation of at least a portion of system resources allocated to a first set of workload tasks in the e-business system from said first set of workload tasks to a second set of workload tasks in response to predicting the overload condition,

executing a query of at least a portion of said first set of workload tasks included in said workload request in response to said first reallocation,

wherein processing of said second set of workload tasks requiring less system resources, wherein said workload tasks are performed by a plurality of different applications under direction of the e-business system; and

if adequate resources in the e-business system become available and if said first set of workload tasks still require processing, causing a second reallocation of system resources to said first set of workload tasks.

11. (Previously Presented) The machine-readable storage according to claim 10, wherein said detecting step further comprises:

receiving said workload request in a text format for providing visualization of a resource requirement of said parameter;

monitoring system parameters within the e-business system; and

analyzing said monitored system parameters to predict when said overload condition occurs in the e-business system.

12. (Previously Presented) The machine-readable storage according to claim 11, wherein said text format is an XML presentation of monitored system parameters comprising CPU utilization, disk I/O and memory utilization.

13. (Previously Presented) A machine-readable storage having stored thereon, a computer program having a plurality of code sections, said code sections executable by a machine for causing the machine to perform the steps of:

receiving a first work request for performing a workload task having an associated workload parameter specifying a resource requirement by at least one application under direction of the e-business system;

receiving HTTP status updates from one or more systems within the e-business system describing an available resource capacity of the one or more systems based on said workload parameter;

updating a status servlet and a core workload driver within an application server using said HTTP status updates;

comparing said resource requirement of said first work request to available system resources to predict if performing said workload task of said first work request is capable of causing a system overload condition in at least one system within the e-business system; and

if performing said workload task of said first work request is capable of causing a system overload condition, transitioning to a second lighter work request, said second lighter work request for performing a different workload task having a lighter workload requiring less system resources, thereby preventing said system overload condition.

14. (Previously Presented) The machine-readable storage according to claim 13, further comprising analyzing an XML representation of said system parameters received within said HTTP request to predict whether performing said workload task of said first work request causes said system overload condition.

15. (Previously Presented) The machine-readable storage according to claim 14, wherein said system parameters comprises CPU utilization, disk I/O and memory utilization.

16. (Previously Presented) The machine-readable storage according to claim 14, further comprising, reporting said system parameters in an XML format under an HTTP request to a workload driver.

17. (Previously Presented) A machine-readable storage having stored thereon, a computer program having a plurality of code sections, said code sections executable by a machine for causing the machine to perform the steps of:

processing a workload task having one or more associated workload parameters in specifying one or more resource requirements performed by at least one application under direction of the e-business system, the workload task assigned in an HTTP request to a workload driver;

monitoring e-business system resources in view of said one or more resource requirements to predict an overload condition in the e-business system while processing said workload task;

allocating processing resources to a lighter workload task when said workload driver predicts a system overload condition caused by said processed workload task during said monitoring step; and,

executing a query of at least a portion of said lighter workload task in response to said first reallocation for acquiring a portion of data requested by said HTTP request,

if said processed workload task still requires processing, reporting a result of said executing a query and responding with a message to make a subsequent request to acquire a remainder of data made available by said processing followed by transitioning

Appln. No. 09/919,439
Amendment dated September 5, 2006
Reply to Office Action of June 2, 2005
Docket No. BOC-2000-0079 (214)

to said processed workload task from said lighter workload task upon availability of adequate processing resources.